

?s pn=jp 2003084984
S3 1 PN=JP 2003084984
?t s3/5

3/5/1
DIALOG(R)File 347:JAPIO
(c) 2005 JPO & JAPIO. All rts. reserv.

07591140 **Image available**
INFORMATION PROCESSOR, INFORMATION PROCESSING METHOD, CONTROL PROGRAM, AND
RECORDING MEDIUM STORING CONTROL PROGRAM

PUB. NO.: 2003-084984 [*JP 2003084984* A]
PUBLISHED: March 20, 2003 (20030320)
INVENTOR(s): IIDA MITSUNORI
APPLICANT(s): CANON INC
APPL. NO.: 2001-276441 [JP 2001276441]
FILED: September 12, 2001 (20010912)
INTL CLASS: G06F-009/445; G06F-003/12; G06F-013/10

ABSTRACT

PROBLEM TO BE SOLVED: To provide a mechanism capable of installing a program while retaining the consistency of another existing control program in a case of using a plug-and-play installer.

SOLUTION: This information processor having a plug-and-play installer is provided with determination means determining whether or not the second control program corresponding to an external processor stored in the information processor becomes inconsistent, when the first control program is installed by program introduction means.

COPYRIGHT: (C)2003, JPO

(19) 日本国特許庁 (J·P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2003-84984

(P2003-84984A)

(43)公開日 平成15年3月20日(2003.3.20)

(51) Int.Cl. 9/445
G 0 6 F 3/12
13/10 3 3 0

F I		テ-マ-コ-ト(参考)
G 0 6 F	3/12	C 5 B 0 1 4
	13/10	3 3 0 B 5 B 0 2 1
	9/06	6 1 0 B 5 B 0 7 6
		6 1 0 L
		6 5 0 B

審査請求 有 請求項の数28 OL (全 19 頁)

(21) 出願番号 特願2001-276441(P2001-276441)

(71) 出圖人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(22)出願日 平成13年9月12日(2001.9.12)

(72)発明者 飯田 光則

東京都大田区下丸子3丁目30番2号キヤノン株式会社内

(74) 代理人 100090538

弁理士 西山 廉三 (外1名)

Fターム(参考) SB014 EB03 FA12 FA15

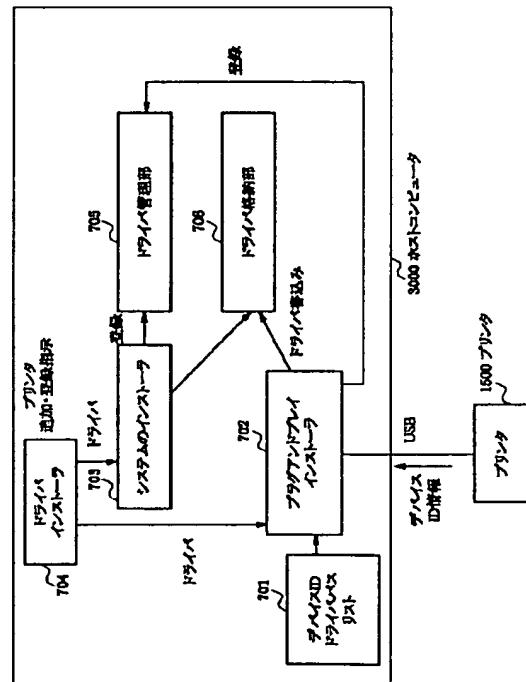
5B021 AA01 BB01 0006

58076 AAD6 ACD7 BB17

(54)【発明の名称】 情報処理装置、及び、情報処理方法、及び、制御プログラム、及び、制御プログラムを記憶した記憶媒体

【課題】 プラグアンドプレイインストーラを用いる場合であっても、他の既存の制御プログラムが整合性を保つようにインストールすることができる仕組みを提供する。

【解決手段】 プラグアンドプレイインストーラを有する情報処理装置において、前記プログラム導入手段により前記第一の制御プログラムがインストールされると、既に前記情報処理装置に記憶された前記外部処理装置に対応する第二の制御プログラムが不整合な状態となるか否かを判断する判断手段を設ける。



【特許請求の範囲】

【請求項 1】 外部処理装置が接続されたことに応答して、外部処理装置の識別情報を受信し、前記外部処理装置に対応する第一の制御プログラムを情報処理装置にインストールする、プログラム導入手手段を有する情報処理装置において、

前記プログラム導入手手段により前記第一の制御プログラムがインストールされると、既に前記情報処理装置に記憶された前記外部処理装置に対応する第二の制御プログラムが不整合な状態となるか否かを判断する判断手段と、

前記情報処理装置のインストール動作を前記判断手段の判断結果に応じて制御する制御手段と、を有することを特徴とする情報処理装置。

【請求項 2】 前記判断手段は、前記プログラム導入手手段がプログラムを導入する前に、前記プログラム導入手手段により前記第一の制御プログラムがインストールされると既に前記情報処理装置に記憶された前記外部処理装置に対応する第二の制御プログラムが不整合な状態となるか否かを判断することを特徴とする請求項 1 に記載の情報処理装置。

【請求項 3】 前記制御手段は、前記判断手段による判断結果を示す情報を報知せるように情報処理装置のインストール動作を制御することを特徴とする請求項 1 に記載の情報処理装置。

【請求項 4】 前記判断手段による判断結果を示す情報とは、前記判断手段により前記第一の制御プログラムがインストールされると前記第二の制御プログラムが不整合な状態になると判断された場合に、前記第一の制御プログラムがインストールされると、前記第二の制御プログラムが不整合な状態になると示す情報であることを特徴とする請求項 3 に記載の情報処理装置。

【請求項 5】 前記判断手段による判断結果を示す情報とは、前記プログラム導入手手段による前記第一の制御プログラムのインストールが開始される前に、第二の制御プログラムの更新を行うべき旨を示す情報であり、

さらに、前記プログラム導入手手段は、前記判断手段により、前記第一の制御プログラムがインストールされると、前記第二の制御プログラムが不整合な状態とはならないと判断された場合に、前記プログラム導入手手段により第一の制御プログラムをインストールさせることを特徴とする請求項 3 に記載情報処理装置。

【請求項 6】 前記制御手段は、前記判断手段により前記第一の制御プログラムがインストールされても、前記第二の制御プログラムが不整合な状態とはならないと判断された場合には、前記プログラム導入手手段により前記情報処理装置に第一の制御プログラムをインストールを開始させ、一方、前記判断手段により、前記第一の制御プログラムがインストールされると、前記第二の制御プログラムが不整合な状態になると判断された場合には、

前記プログラム導入手手段による前記第一の制御プログラムのインストールを開始させないように前記情報処理装置のインストール動作を制御することを特徴とする請求項 1 に記載の情報処理装置。

【請求項 7】 前記第二の制御プログラムの不整合な状態とは、前記第一の制御プログラムがインストールされることにより、既にインストールされている前記第二の制御プログラムの一部が、上書きされてしまう場合であり、

前記判断手段は、前記プログラムの一部が上書きされてしまうか否かを、前記第一の制御プログラムのバージョンを示す情報と、前記第二の制御プログラムのバージョンを示す情報を比較することにより判断することを特徴とする、請求項 1 に記載の情報処理装置。

【請求項 8】 前記判断手段は、各外部処理装置に制御する各制御プログラムにおいて共有されるモジュール情報が、前記第一の制御プログラムがインストールされることにより上書きされるか否かを判断し、前記共有されるモジュールが上書きされる場合には、前記第二の制御プログラムが不整合な状態になると判断することを特徴とする請求項 1 に記載の情報処理装置。

【請求項 9】 前記情報処理装置は、さらに、プログラム更新手段を有し、

前記判断手段により前記第一の制御プログラムがインストールされると前記情報処理装置に記憶された制御プログラムの整合性がとれなくなると判断された場合に、前記制御手段は、前記プログラム導入手手段による前記第一の制御プログラムのインストールとは別処理として、前記プログラム更新手段による前記第二の制御プログラムの更新処理を実行させるように制御することを特徴とする請求項 1 に記載の情報処理装置。

【請求項 10】 前記判断手段は、前記情報処理装置は、前記第一の制御プログラムがインストールされると前記情報処理装置に記憶されている制御プログラムの整合性がとれなくなる複数の制御プログラムを検出し、前記情報処理装置は、前記検出された前記情報処理装置に記憶されている複数の制御プログラムをまとめて更新させることを特徴とする請求項 1 に記載の情報処理装置。

【請求項 11】 前記判断手段は、前記外部処理装置の接続に応答して動作し、

前記前記第一の制御プログラムがインストールされると既に前記情報処理装置に記憶された前記外部処理装置の第二の制御プログラムが不整合な状態となるか否かを、前記外部処理装置から受信した前記外部処理装置の識別情報を用いて判断することを特徴とする、請求項 1 に記載の情報処理装置。

【請求項 12】 ネットワークを介して外部処理装置接続されたことに応答して外部処理装置の識別情報を受信することを特徴とする請求項 1 に記載の情報処理装置。

【請求項 13】 外部処理装置が接続されたことに応答して、外部処理装置の識別情報を受信し、前記外部処理装置に対応する第一の制御プログラムを、情報処理装置にインストールするプログラム導入手段を有する情報処理装置において実行される制御プログラムであって、前記プログラム導入手段により前記第一の制御プログラムがインストールされると、既に前記情報処理装置に記憶された前記外部処理装置に対応する第二の制御プログラムが不整合な状態となるか否かを判断する判断工程と、

前記判断工程の判断結果に応じて、前記情報処理装置のインストール動作を制御する制御工程と、を有する前記情報処理装置において実行される制御プログラム。

【請求項 14】 前記プログラム導入手段により前記第一の制御プログラムのインストールが開始される前に、前記判断工程を実行することを特徴とする請求項 13 に記載の制御プログラム。

【請求項 15】 前記制御工程は、前記判断工程による判断結果を示す情報を報知するように前記情報処理装置のインストール動作を制御する工程であることを特徴とする請求項 13 に記載の制御プログラム。

【請求項 16】 前記判断工程における判断結果を示す情報とは、前記判断工程により前記第一の制御プログラムがインストールされると、前記第二の制御プログラムが不整合な状態になると判断された場合には、前記第一の制御プログラムがインストールされると、前記第二の制御プログラムが不整合な状態になると示す情報を有することを特徴とする請求項 15 に記載の制御プログラム。

【請求項 17】 前記判断工程による判断結果を示す情報とは、前記プログラム導入手段による前記第一の制御プログラムのインストールが開始される前に、第二の制御プログラムの更新を行うべき旨を示す情報であり、さらに、前記プログラム導入手段は、前記判断工程において、前記第一の制御プログラムがインストールされると、前記第二の制御プログラムが不整合な状態とはならないと判断された場合に、前記プログラム導入手段により第一の制御プログラムをインストールさせることを特徴とする請求項 15 に記載の制御プログラム。

【請求項 18】 前記制御工程においては、前記判断工程により前記第一の制御プログラムがインストールされると、前記第二の制御プログラムが不整合な状態とはならないと判断された場合には、前記プログラム導入手段による第一の制御プログラムのインストールを開始させ、

一方、前記判断工程において、前記第一の制御プログラムがインストールされると、前記第二の制御プログラムが不整合な状態になると判断された場合には、前記プログラム導入手段による前記第一の制御プログラムのインストールを開始させないように前記情報処理装置のイン

ストール動作を制御する制御工程を前記情報処理装置に実行させることを特徴とする請求項 13 に記載の制御プログラム。

【請求項 19】 前記第二の制御プログラムの不整合な状態とは、前記第一の制御プログラムを組み込むことにより、既にインストールされている前記第二の制御プログラムの一部が、上書きされてしまう場合であり、前記判断工程においては、前記プログラムの一部が上書きされてしまうか否かは、前記第一の制御プログラムのバージョンを示す情報と、前記第二の制御プログラムのバージョンを示す情報を比較することにより判断されることを特徴とする、請求項 13 に記載の制御プログラム。

【請求項 20】 前記判断工程においては、各外部処理装置に制御する各制御プログラムにおいて共有されるモジュール情報が、前記第一の制御プログラムをインストールすることにより上書きされるか否かを判断され、前記共有されるモジュールが上書きされる場合には、前記第二の制御プログラムが不整合な状態になると判断されることを特徴とする請求項 13 に記載の制御プログラム。

【請求項 21】 前記情報処理装置は、さらに、プログラム更新手段を有し、

前記制御プログラムは、前記判断工程により前記第一の制御プログラムがインストールされると前記情報処理装置に記憶された制御プログラムの整合性がとれなくなると判断された場合に、前記制御工程においては、前記プログラム導入手段による前記第一の制御プログラムのインストールとは別処理として、前記プログラム更新手段による前記第二の制御プログラムの更新処理を情報処理装置において実行させることを特徴とする請求項 13 に記載の制御プログラム。

【請求項 22】 前記判断工程において、前記情報処理装置は、前記第一の制御プログラムがインストールされると前記情報処理装置に記憶されている制御プログラムの整合性がとれなくなる複数の制御プログラムを検出させ、

前記検出された前記情報処理装置に記憶されている複数の制御プログラムをまとめて更新させるよう情報処理装置を実行させることを特徴とする請求項 13 に記載の制御プログラム。

【請求項 23】 前記判断工程は、前記外部処理装置の接続に応答して実行され、前記前記第一の制御プログラムがインストールされると既に前記情報処理装置に記憶された前記外部処理装置の第二の制御プログラムが不整合な状態となるか否かは、前記外部処理装置から受信した前記外部処理装置の識別情報により判断されることを特徴とする請求項 13 に記載の制御プログラム。

【請求項 24】 ネットワークを介して外部処理装置が接続されたことに応答して外部処理装置の識別情報を受

信することを特徴とする情報処理装置において実行されることを特徴とする請求項13に記載の制御プログラム。

【請求項25】 請求項13乃至24のいずれかに記載の制御プログラムを格納したコンピュータ読取り可能な記憶媒体。

【請求項26】 外部処理装置が接続されたことに応答して、外部処理装置の識別情報を受信し、前記外部処理装置に対応する第一の制御プログラムを、情報処理装置にインストールさせるプログラム導入手段を有する情報処理装置における方法であって、

前記第一の制御プログラムがインストールされると、既に前記情報処理装置に記憶された前記外部処理装置に対応する第二の制御プログラムが不整合な状態となるか否かを、前記第一の制御プログラムのインストールを開始する前に判断する判断工程と、

前記プログラム導入工程による前記第一の制御プログラムのインストールが開始される前に、前記判断工程による判断結果を示す情報を報知する報知工程と、を有する情報処理方法。

【請求項27】 外部処理装置が接続されたことに応答して、外部処理装置の識別情報を受信し、前記外部処理装置に対応する第一の制御プログラムを、情報処理装置にインストールするプログラム導入手段を有する情報処理装置における方法であって、

前記前記第一の制御プログラムがインストールされると、既に前記情報処理装置に記憶された前記外部処理装置の第二の制御プログラムが不整合な状態となるか否かを判断する判断工程と、

前記判断工程により前記第一の制御プログラムがインストールされると、前記第二の制御プログラムが不整合な状態とはならないと判断された場合には、前記プログラム導入工程により前記情報処理装置に第一の制御プログラムをインストールさせ、一方、前記判断工程により、前記第一の制御プログラムがインストールされると、前記第二の制御プログラムが不整合な状態になると判断された場合には、前記プログラム導入手段によって前記第一の制御プログラムのインストールを開始させないように制御する制御工程を有することを特徴とする情報処理方法。

【請求項28】 外部処理装置が接続されたことに応答して、外部処理装置の識別情報を受信し、前記外部処理装置に対応する第一の制御プログラムを情報処理装置にインストールする、プログラム導入手段を有する情報処理装置を制御する制御方法であって前記プログラム導入手段により前記第一の制御プログラムがインストールされると、既に前記情報処理装置に記憶された前記外部処理装置に対応する第二の制御プログラムが不整合な状態となるか否かを判断する判断工程と、

前記情報処理装置のインストール動作を前記判断手段の

判断結果に応じて制御する制御工程と、を有することを特徴とする情報処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、外部処理装置装置に接続すると、接続された外部処理装置に対応した制御ソフトウェアが導入される情報処理装置におけるものである。

【0002】

【従来の技術】 様々な外部処理装置を制御する制御プログラムの好適な一例として、プリンタドライバがある。プリンタドライバは、OS上で動作するアプリケーションから外部処理装置の好適な一例であるプリンタ等の印刷装置に印刷処理を行わせるために、OSにインストールする必要のあるソフトウェアである。このプリンタドライバを、ホストコンピュータに搭載されたOSに対してインストールを行うインストーラが従来から存在する。また、印刷装置からデバイス固有の識別情報をあるデバイスIDを入手し、このデバイスIDを用いてプリンタドライバのインストールを行うコンピュータが開発されている。

【0003】

【発明が解決しようとする課題】 一般に、プリンタドライバは装置毎に異なったソフトウェアであるため、ホストコンピュータに搭載されたOSへのインストールが煩雑であった。

【0004】 そこで、印刷装置がシステムに接続されると、印刷装置からその印刷装置固有の情報を示す情報である識別情報を取得し、対応するプリンタドライバのインストールを行うという機能(以降、プラグアンドプレイ機能と呼ぶ)がOSに追加され、ユーザーの利便性向上が図られてきている。また、上記のように、プラグアンドプレイのインストールを容易に行うことができる、プラグアンドプレイ対応のインストーラ(以下、プラグアンドプレイインストーラと呼ぶ)が提供されている。このプラグアンドプレイインストーラは、通常、システムソフトウェア、例えば、オペレーティングシステム(OS)の一機能として提供される。

【0005】 プラグアンドプレイインストーラにおいては、印刷装置に対応したプリンタドライバの情報をOSが認識できるように準備を行い、その後、ユーザーに印刷装置を接続させるのに応答して、デバイス識別情報をデバイスから受信して、デバイス識別情報に対応するドライバを特定し、プリンタドライバのインストールを行っていた。

【0006】 また、従来、デバイスドライバの開発は、デバイスごとに行われていたため、一つのデバイスドライバ内のすべてのモジュールが対応していた。しかしながら、従来のように、あるデバイスに対して、特定のデバイスドライバの開発を行うと、開発工程が長くなつて

しまい、開発者の負担が大きいものであった。そのため、近年では、開発工程を減らすために、デバイスドライバを共通モジュール部分と、デバイス固有モジュール部分に分けることにより、共通モジュール部分の開発工程を減らす工夫がなされている。それにより、特定のデバイスに対応するデバイスドライバの開発では、デバイス固有モジュール部分の開発工程ですみ、開発者の負担が減るため、近年は共通モジュールを有するデバイスドライバが一般的になってきている。

【0007】しかしながら、共通モジュールを有するデバイスドライバにおいて、プラグアンドプレイによってデバイスドライバのインストールを行うと、次のような問題が生ずる。プラグアンドプレイインストーラは、外部処理装置等、各種機器の識別情報であるデバイスIDに基づいて、インストールする制御プログラムを特定し、特定されたデバイスドライバをそのままインストールしてしまう。このため、ユーザの意図に反して、他の既存の制御プログラムを上書きしてしまう。ここで、デバイスドライバの共通モジュールは、デバイスのラインナップが増えることに応じて、少なからず、拡張が行われているため、このために共通モジュールの上書き処理を行うと、バージョンの古いデバイスドライバが既にシステムにインストールされている場合に、ソフトウェアに不整合が生ずる場合があった。

【0008】本願の実施形態においては、プラグアンドプレイインストーラを用いる場合であっても、他の既存の制御プログラムが整合性を保つようにインストールすることができる仕組みを提供することが課題となる。

【0009】

【課題を解決するための手段】本発明は、上記の課題を解決するためになされたものであり、本願の実施形態においては、プラグアンドプレイインストーラを用いて新たに制御プログラムのインストールを行う場合であっても、他の既存の制御プログラムが、インストールされるプログラムによって意図しない上書きをされることなく、整合性を保つことができる仕組みを提供することを目的とする。これらの目的を達成するための構成として、以下の手段を提供する。

【0010】請求項1に記載の発明は、外部処理装置が接続されたことに応答して、外部処理装置の識別情報を受信し、前記外部処理装置に対応する第一の制御プログラムを情報処理装置にインストールする、プログラム導入手段を有する情報処理装置において、前記プログラム導入手段により前記第一の制御プログラムがインストールされると、既に前記情報処理装置に記憶された前記外部処理装置に対応する第二の制御プログラムが不整合な状態となるか否かを判断する判断手段と、前記情報処理装置のインストール動作を前記判断手段の判断結果に応じて制御することを特徴とする。

【0011】請求項2に記載の発明は、前記判断手段

は、前記プログラム導入手手段がプログラムを導入する前に、前記プログラム導入手手段により前記第一の制御プログラムがインストールされると既に前記情報処理装置に記憶された前記外部処理装置に対応する第二の制御プログラムが不整合な状態となるか否かを判断することを特徴とする。

【0012】請求項3に記載の発明は、前記制御手段は、前記判断手段による判断結果を示す情報を報知させるように情報処理装置のインストール動作を制御することを特徴とする。

【0013】請求項4に記載の発明は、前記判断手段による判断結果を示す情報とは、前記判断手段により前記第一の制御プログラムがインストールされると前記第二の制御プログラムが不整合な状態になると判断された場合に、前記第一の制御プログラムがインストールされると、前記第二の制御プログラムが不整合な状態になることを示す情報であることを特徴とする。

【0014】請求項5に記載の発明は、前記判断手段による判断結果を示す情報とは、前記プログラム導入手手段による前記第一の制御プログラムのインストールの開始される前に、第二の制御プログラムの更新を行うべき旨を示す情報であり、さらに、前記プログラム導入手手段は、前記判断手段により、前記第一の制御プログラムがインストールされると、前記第二の制御プログラムが不整合な状態とはならないと判断された場合に、前記プログラム導入手手段により第一の制御プログラムをインストールさせることを特徴とする。

【0015】請求項6に記載の発明は、前記制御手段は、前記判断手段により前記第一の制御プログラムがインストールされても、前記第二の制御プログラムが不整合な状態とはならないと判断された場合には、前記プログラム導入手手段により前記情報処理装置に第一の制御プログラムのインストールを開始させ、一方、前記判断手段により、前記第一の制御プログラムがインストールされると、前記第二の制御プログラムが不整合な状態になると判断された場合には、前記プログラム導入手手段による前記第一の制御プログラムのインストールを開始させないように前記情報処理装置のインストール動作を制御することを特徴とする。

【0016】請求項7に記載の発明は、前記第二の制御プログラムの不整合な状態とは、前記第一の制御プログラムがインストールされることにより、既にインストールされている前記第二の制御プログラムの一部が、上書きされてしまう場合であり、前記判断手段は、前記プログラムの一部が上書きされてしまうか否かを、前記第一の制御プログラムのバージョンを示す情報と、前記第二の制御プログラムのバージョンを示す情報を比較することにより判断することを特徴とする。

【0017】請求項8に記載の発明は、前記判断手段は、各外部処理装置に制御する各制御プログラムにおい

て共有されるモジュール情報が、前記第一の制御プログラムがインストールされることにより上書きされるか否かを判断し、前記共有されるモジュールが上書きされる場合には、前記第二の制御プログラムが不整合な状態になると判断すること特徴とする。

【0018】請求項9に記載の発明は、前記情報処理装置は、さらに、プログラム更新手段を有し、前記判断手段により前記第一の制御プログラムがインストールされると前記情報処理装置に記憶された制御プログラムの整合性がとれなくなると判断された場合に、前記制御手段は、前記プログラム導入手段による前記第一の制御プログラムのインストールとは別処理として、前記更新手段による前記第二の制御プログラムの更新処理を実行させるように制御することを特徴とする。

【0019】請求項10に記載の発明は、前記判断手段は、前記情報処理装置は、前記第一の制御プログラムをインストールすると前記情報処理装置に記憶されている制御プログラムの整合性がとれなくなる複数の制御プログラムを検出し、前記情報処理装置は、前記検出された前記情報処理装置に記憶されている複数の制御プログラムをまとめて更新させることを特徴とする。

【0020】請求項11に記載の発明は、前記判断手段は、前記外部処理装置の接続に応答して動作し、前記前記第一の制御プログラムがインストールされると既に前記情報処理装置に記憶された前記外部処理装置の第二の制御プログラムが不整合な状態となるか否かを、前記外部処理装置から受信した前記外部処理装置の識別情報を用いて判断することを特徴とする。

【0021】請求項12に記載の発明は、ネットワークを介して外部処理装置接続されたことに応答して外部処理装置の識別情報を受信することを特徴とする。

【0022】請求項26に記載の発明は、外部処理装置が接続されたことに応答して、外部処理装置の識別情報を受信し、前記外部処理装置に対応する第一の制御プログラムを、情報処理装置にインストールさせるプログラム導入手段を有する情報処理装置における方法であって、前記第一の制御プログラムがインストールされると、既に前記情報処理装置に記憶された前記外部処理装置に対応する第二の制御プログラムが不整合な状態となるか否かを、前記第一の制御プログラムのインストールを開始する前に判断する判断工程と、前記プログラム導入手工程による前記第一の制御プログラムのインストールが開始される前に、前記判断工程による判断結果を示す情報を報知する報知工程とを有することを特徴とする。

【0023】請求項27に記載の発明は、外部処理装置が接続されたことに応答して、外部処理装置の識別情報を受信し、前記外部処理装置に対応する第一の制御プログラムを、情報処理装置にインストールするプログラム導入手段を有する情報処理装置における方法であって、前記前記第一の制御プログラムがインストールされる

と、既に前記情報処理装置に記憶された前記外部処理装置の第二の制御プログラムが不整合な状態となるか否かを判断する判断工程と、前記判断工程により前記第一の制御プログラムがインストールされると、前記第二の制御プログラムが不整合な状態とはならないと判断された場合には、前記プログラム導入手工程により前記情報処理装置に第一の制御プログラムをインストールさせ、一方、前記判断工程により、前記第一の制御プログラムをインストールすると、前記第二の制御プログラムが不整合な状態になると判断された場合には、前記プログラム導入手段によって前記第一の制御プログラムのインストールを開始させないように制御する制御工程を有することを特徴とする。

【0024】なお、請求項13乃至請求項25に記載の発明は、請求項1乃至11記載の発明にそれぞれ対応したプログラム並びにプログラムを格納した記憶媒体の発明であり、請求項28に記載の発明は、請求項1に対応する方法の発明であるので省略する。

【0025】

【発明の実施の形態】以下、本発明を適用するのに好適である実施例について説明を行う。制御プログラムには、外部処理装置を制御するためのソフトウェアであるデバイスドライバが含まれる。また、デバイスドライバには、プリンタ制御するためのソフトウェアであるプリンタドライバが含まれる。本実施例においては、制御プログラムの好適な一例として、プリンタドライバを取り上げて説明する。プリンタドライバのモジュール構成として、複数の機種で共通のモジュールファイルと、機種毎に異なるモジュールファイルの組み合わせからなっているものがある。このような構成のドライバ(以降、ファミリードライバと呼ぶ)では、ドライバのバージョンアップに伴ってソフトウェアの不具合が修正された場合、共通モジュールファイルに原因のある不具合は、全機種で共通に修正されるという利点がある。しかし、新バージョンのドライバをシステムにインストールした場合、既にシステムに旧バージョンのドライバによって他の機種のインストールが行われていると、インストール済みであった機種のプリンタドライバのモジュール構成は、共通モジュールのみが新バージョンのものに置き換わることとなり、そのことによって不具合が発生する場合があるという問題がある。ここで、インストールとは、所望のプログラムを装置に組み込んで利用可能な状態とすることである。また、このようなプログラムのインストールを行うプログラム導入手段であるソフトウェアを、インストーラと呼ぶ。そこで、本願発明の一つの実施例においては、既にシステムに旧バージョンのドライバによって他の機種のインストールが行われていても、インストール済みであった機種のプリンタドライバのモジュール構成を正しく更新し、ソフトウェア更新により不具合が発生することを防ぐことが本実施例の課題

の一つとなる。

【0026】(第一実施形態) 図1は本発明の実施形態における制御プログラムのインストールシステムの実施例を示すプリンタ制御システムの構成を示す図である。なお、特に断らない限り、本発明の機能が実行されるのであれば、単体の機器であっても、複数の機器からなるシステムであっても、LAN, WAN等のネットワークを介して接続が為され処理が行われるシステムであっても本発明を適用できることは言うまでもない。まず、本実施例の前提となるシステムを説明し、続いて、前提となるシステムの課題を明らかにする。

【0027】図1において、3000は情報処理装置の好適な一例であるホストコンピュータで、ROM3のプログラム用ROMあるいは外部メモリ11に記憶された文書処理プログラム等に基づいて图形、イメージ、文字、表(表計算等を含む)等が混在した文書処理を実行するCPU1を備え、システムバス4に接続される各デバイスをCPU1が総括的に制御する。また、このROM3のプログラム用ROMあるいは外部メモリ11には、CPU1を制御するオペレーティングシステムプログラム(以下OS)等を記憶し、ROM3のフォント用ROMあるいは外部メモリ11には上記文書処理の際に使用するフォントデータ等を記憶し、ROM3のデータ用ROMあるいは外部メモリ11には上記文書処理等を行なう際に使用する各種データを記憶する。2はRAMで、CPU1の主メモリ、ワークエリア等として機能する。5はキーボードコントローラ(KBC)で、キーボード9や不図示のポインティングデバイスからのキー入力を制御する。6はCRTコントローラ(CRTC)で、CRTディスプレイ(CRT)10の表示を制御する。7はディスクコントローラ(DKC)で、ブートプログラム、各種のアプリケーション、フォントデータ、ユーザファイル、編集ファイル、外部処理装置の制御プログラムの好適な一例であるデバイスドライバ等を記憶するハードディスク(HD)、フロッピー(登録商標)ディスク(FD)等の外部メモリ11とのアクセスを制御する。本実施例では、デバイスドライバの好適な一例であるプリンタドライバを例に説明する。本実施例におけるプリンタドライバとは、プリンタ1500を制御するプリンタ制御コマンド生成プログラムのことである。8はプリンタコントローラ(PTC)で、所定の双方向性インターフェース(インターフェース)21を介して外部処理装置の一例として好適な画像形成装置であるプリンタ1500に接続されて、プリンタ1500との通信制御処理を実行する。外部処理装置はプリンタ以外の、例えばスキャナやデジタルカメラ、デジタルビデオカメラなどの周辺装置であってもよく、画像形成装置は、プリンタ、ファクシミリ、複写機、並びにこれらの複合機を含む。なお、CPU1は、例えばRAM2上に設定された表示情報RAMへのアウトラインフォントの

展開(ラスタライズ)処理を実行し、表示手段であるCRT10上でのWYSIWYGを可能としている。また、CPU1は、CRT10上の不図示のマウスカーソル等で指示されたコマンドに基づいて登録された種々のウインドウを開き、種々のデータ処理を実行する。ユーザーは印刷を実行する際、印刷の設定に関するウインドウを開き、プリンタの設定や、印刷モードの選択を含むプリンタドライバに対する印刷処理方法の設定を行える。

【0028】プリンタ1500において、12はプリンタCPUで、ROM13のプログラム用ROMに記憶された制御プログラムあるいは外部メモリ14に記憶された制御プログラムに基づいてシステムバス15に接続される印刷部(プリンタエンジン)17に出力情報としての画像信号を出力する。また、このROM13のプログラムROMには、CPU12の制御プログラム等を記憶する。ROM13のフォント用ROMには上記出力情報を生成する際に使用するフォントデータ等を記憶し、ROM13のデータ用ROMにはハードディスク等の外部メモリ14がないプリンタの場合には、主としてホストコンピュータ上で利用される情報を記憶している。CPU12は入力部18を介してホストコンピュータとの通信処理が可能となっており、プリンタ内の情報をホストコンピュータ3000に通知可能に構成されている。19はCPU12の主メモリ、ワークエリア等として機能するRAMで、図示しない増設ポートに接続されるオプションRAMによりメモリ容量を拡張することができるよう構成されている。なお、RAM19は、出力情報展開領域、環境データ格納領域、NVRAM等に用いられる。前述したハードディスク(HD)、ICカード等の外部メモリ14は、メモリコントローラ(MC)20によりアクセスを制御される。外部メモリ14は、オプションとして接続され、フォントデータ、エミュレーションプログラム、フォームデータ等を記憶する。また、18は前述した操作パネルで操作のためのスイッチおよびLED表示器等が配されている。また、前述した外部メモリは1個に限らず、少なくとも1個以上備え、内蔵フォントに加えてオプションフォントカード、言語系の異なるプリンタ制御言語を解釈するプログラムを格納した外部メモリを複数接続できるように構成されてもよい。さらに、図示しないNVRAMを有し、操作パネル1501からのプリンタモード設定情報を記憶するようにしてもよい。

【0029】図2は、プリンタ等の印刷装置が直接接続されているか、あるいはネットワーク経由で接続されている情報処理装置であるホストコンピュータ3000における典型的な印刷処理の例を示す図である。アプリケーション201、グラフィックエンジン202、プリンタドライバ203、およびシステムスプーラ204は、外部メモリ11に保存されたファイルとして存在し、実行される場合にOSやそのモジュールを利用するモジュ

ールによってRAM2にロードされ実行されるプログラムモジュールである。また、アプリケーション201およびプリンタドライバ203は、外部メモリ11のFDやCD-ROM(図示省略)、あるいはネットワーク(図示省略)を経由して外部ディスク11のHDに追加することが可能となっている。外部メモリ11に保存されているアプリケーション201はRAM2にロードされて実行されるが、このアプリケーション201からプリンタ1500に対して印刷を行う際には、同様にRAM2にロードされ実行可能となっているグラフィックエンジン202を利用して出力(描画)を行う。グラフィックエンジン202は印刷装置ごとに用意されたプリンタドライバ203を同様に外部メモリ11からRAM2にロードし、アプリケーション210の出力を、プリンタドライバ203を用いてプリンタの制御コマンドに変換する。変換されたプリンタ制御コマンドはOSによってRAM2にロードされたシステムスプーラ204を経てインターフェース211経由でプリンタ1500へ出力される仕組みとなっている。

【0030】図7は、本発明の実施例におけるホストコンピュータ3000内のソフトウェア構成を示す図である。701は、デバイスID並びにドライバリストの組からなるデータベースである。702～706は、プログラムモジュールであり、ホストコンピュータ3000内の図1の外部メモリ11に記憶されており、必要に応じてRAM2に読み出されてCPU1によって制御される。ここで、デバイスID情報とは、周辺装置(例えば、プリンタ1500)を識別するための情報である。ここでいう、ドライバリストとは、特定のデバイスIDに対応する周辺装置を利用可能にする制御プログラムの好適な一例であるデバイスドライバ(本実施例ではプリンタドライバ)の外部メモリ11における格納場所(パス)のことである。図7においては、ドライバの格納場所は、外部HDDとなっているが、ネットワークを介して接続された他の外部サーバ装置内のメモリ(図示省略)内であってもよい。

【0031】まず、従来のインストーラ703(システムインストーラ703)について説明する。本実施例におけるプラグアンドプレイ機能が開発される前から使用されているシステムインストーラ703は、ここでは、OSが提供する機能である。704は、ドライバインストーラである。本実施形態におけるホストコンピュータを制御する制御手段の好適な一例であるドライバインストーラ704は、アプリケーションであり、外部メモリ11又は他のサーバ装置(図示省略)から、ユーザーによるデバイスドライバの選択などの指示に基づいて、デバイスドライバをドライバ格納部706に組み込む(インストールする)ようシステムインストーラ703に指示する機能を有するモジュールである。ここで、システムインストーラが、システムインストーラ703にデバイス

ドライバの格納場所を伝えてインストールを指示すると、システムインストーラ703は、デバイスドライバをドライバ格納部706にインストールし、ドライバ管理部705にデバイスドライバを使用する外部処理装置の登録を行う。ドライバ管理部705は、レジストリとも呼ばれ、OSに対して外部処理装置を認識させるべく、外部処理装置の情報を登録するOS内の場所である。ここでは具体的には、デバイスドライバのインストールをシステムインストーラ703を介して行う場合、ユーザーは、システムインストーラ703に対して、ドライバインストーラ704を介して、デバイスドライバの追加・更新、並びに、外部処理装置のOSの一機能である外ドライバ登録部への登録などの詳細な指示を行うことが出来る。ここでは、外部処理装置登録部の好適な一例であるドライバ登録部へのプリンタの登録により、ドライバ格納部705に格納されたデバイスドライバと、このデバイスドライバを使用するプリンタがOSに登録される。ドライバ格納部706は使用可能としたい外部処理装置に対応するデバイスドライバが組み込まれる外部メモリ11の一部である。

【0032】次に、プラグアンドプレイに対応した、プログラム導入手段であるプラグアンドプレイインストーラ702について説明する。プラグアンドプレイインストーラ702は、外部処理装置の好適な一例であるプリンタ1500がUSBを介して接続されたことに応答して、プリンタ1500から機器の識別情報の好適な一例であるデバイスID情報を受信し、デバイスID・ドライバリスト701から受信したデバイスID情報に対応するプリンタドライバの格納場所を特定する。新たにインストール又は更新しようとするプリンタドライバが、第一の制御プログラムの好適な一例である。なお、本実施例では機器の識別情報を送受信するための方法の一例として、USBを用いているが、外部処理装置の識別情報であるデバイスID情報は、構造化言語であるextensible Markup Language(XML)等を含むメッセージの形式等で、より上位の層でやりとりされてもよい。次に、プラグアンドプレイインストーラは、第一の制御プログラムの好適な一例である装置外のサーバ(図示省略)又はホストコンピュータ3000の有する外部メモリ14に記憶されたプリンタドライバを、ドライバ格納部706にインストールする。同時に、自動的にドライバ管理部にデバイスID情報に基づいてプリンタを登録する。ここで、従来のシステムインストーラ703は、その上位にあるドライバインストーラ704を介してユーザーから追加又は更新するプリンタドライバの選択指示を受けることが出来たため、ユーザーが、システム内のプリンタドライバの整合性を保つべく、新たにプリンタドライバをインストールする途中で、他にも更新しておきたいプリンタドライバを選択して更新することができた。しかし、プラグアンドプレイインストーラ702

は、ユーザの入力を介して起動されるものではなく、受信したデバイスIDに基づいて、第一の制御プログラムの一例である更新・追加するべきプリンタドライバを自動的に選択してしまう。一方、プラグアンドプレイインストーラ702は、プリンタドライバを新たにインストールする前に、ユーザが追加又は更新するドライバを選択指示する手段を提供していないため、次のような問題が生ずる。すなわち、プリンタドライバの追加・更新時に際して、ユーザは、共有モジュールを有する既存のプリンタドライバに対してどのような変更が行われようとしているかを事前に知ることができず、既存のプリンタドライバの共有モジュールを変更することを望まない場合でも、ユーザの意図に反して既存の共有モジュールを有するプリンタドライバが上書きされてしまうという問題が生ずる。また、プラグアンドプレイインストーラ702は、受信したデバイスIDに基づいてインストールすべきドライバが一律に選択されてしまうため、従来のシステムインストーラ703とは異なり、ドライバインストーラ704を介して、既存のプリンタドライバの追加・更新、並びにプリンタのドライバ管理部705への登録などの詳細な指示を行うことが出来ないという問題がある。具体的には、既に他の機種のプリンタドライバが入っているコンピュータに対して、プリンタAを接続すると、自動的或いは半自動的にプリンタAのプリンタドライバをドライバ格納部706に書き込むことになり、他のプリンタドライバの共有モジュールなどを上書きしてしまう問題などについては考慮されていない問題である。そこで、本願発明の実施例においては、プラグアンドプレイインストーラ702を用いるインストールを行う場合であっても、プリンタドライバの整合性を保ちつつ、プリンタドライバの更新または追加を行うことができる仕組みを提供することが課題となる。続いて、図3以降において、これらの課題を解決する方法を詳細に述べる。

【0033】図3は、ドライバを構成するモジュールの内容、並びに、新たにドライバをインストールする際の処理を示す流れ図である。図3にはプリンタドライバ(バージョン1.0)並びにプリンタドライバ(バージョン2.0)の構成図が示されている。本図に示すプリンタドライバは、プリンタAおよびプリンタBをサポートしており、バージョン1.0においては、プリンタA用のプリンタドライバは、共通モジュールの1～3と機種別モジュールのモジュールAで構成され、プリンタB用のプリンタドライバは、共通モジュール1～3と機種別モジュールのモジュールBで構成されている。また、バージョン2.0においては、バージョン1.0でのモジュールがすべて更新され(更新されたことを示すため、'.'(ダッシュ)をついている)ており、また、両プリンタに共通するモジュールについては、共通モジュール4が一つ追加となっている。

【0034】今、図3の①に示すように、プリンタAに対応するドライバが、ホストコンピュータ3000にインストールされている状態を考える。ここで、従来のプラグアンドプレイインストーラによってプリンタBを追加すると、システム内のドライバは、図3の②の状態になってしまう。すなわち、プリンタBのバージョン2.0のプリンタドライバは前述した通り、共通モジュール1'～3'並びに共通モジュール4、並びに、プリンタB用モジュールB'によって構成されている。従って、プリンタBのプリンタドライバを追加すると、プリンタA用の共通モジュール1'～3'がバージョン2.0として上書きされるにも関わらず、プリンタA用モジュールAがバージョン1.0のままとなる。従って、各共通モジュールのバージョンが一致せず、プリンタドライバAが不整合な状態となっており、その動作に不具合が発生する可能性があるという問題がある。従来のプラグアンドプレイ非対応のインストーラでは、ここではバージョン番号でドライバモジュールを管理しているが、他にバージョン情報として、数字以外の識別可能な記号・文字列などの識別子によりプログラムを管理していてもよいことは言うまでもない。ここで、制御プログラムの不整合な状態とは、例えば、一つの機種に対応するプリンタドライバの共通モジュールと、各ドライバの機種別モジュールにおいて用意されたインターフェースが異なってしまったため不整合となる状態や、各ドライバの機種別モジュールと共有モジュールにも用意されることのある機種別データなどの不整合な状態がある。本実施例では、好適な一例として、機種別モジュールのバージョン情報によって、不整合な状態となるか否かを調べているが、各モジュール間の関数呼出し・インターフェースの検査プログラム等を用意してチェックしてもよい。本実施例では、好適な例として、できるだけ既存のドライバを最新のプリンタドライバに更新させるような設計になっている。他の実施例としては、例えば、ドライバインストーラに外部から特定の範囲のバージョンを設定出来るようにし、前記特定の範囲のバージョンの既存のドライバを、現在導入しようとしているバージョンのプリンタドライバに更新させる仕組みにしてもよく、既存のプリンタドライバを必ずしも最新のプリンタドライバに更新する必要はない。

【0035】図4に示すのが、本実施例におけるドライバインストーラ704を起動した際に表示されるプリンタドライバのインストール方法、すなわち、インストール方法を選択するためのユーザインターフェース(U1)例を示す図である。以下、図3並びに図4を用いてプラグアンドプレイインストーラにおける問題の解決方法の一例を示す。

【0036】制御プログラムのインストールとは、自装置内に予めダウンロードされた外部制御装置の制御プログラムの好適な一例であるプリンタドライバを、自装置

内のシステムが利用可能な状態に組むことをいい、例えば、ドライバを、ホストコンピュータ3000に搭載されたOSによるロードが可能な状態として、ドライバ格納部706に組み込んだ状態とすることを含む。もちろん、外部サーバ装置（図示省略）に保持されたプリンタドライバを、ホストコンピュータがダウンロードして組み込むネットワークインストールも、インストールの好適な一例である。

【0037】例えば、今、ホストコンピュータ3000のシステム内のドライバが、図3の①の状態になっているとする。まず、ユーザはプリンタドライバインストーラを起動し、プラグアンドプレイ機能によってプリンタドライバをインストールするために、「USB接続でドライバをインストールする」をチェックする（図2の①）。その後、「次へ」ボタンを押下すると、後に図5並びに図6を用いて詳述するが、プリンタBを制御するプリンタドライバをインストールすると、既にホストコンピュータに記憶されているプリンタAを制御するプリンタドライバが、不整合な状態となるか否かを判断する判断処理（図5のS510及びS511、又は、図6のS610又はS611）が行われ、プラグアンドプレイインストーラ702によるプリンタBのプリンタドライバのインストールが開始される前に、ドライバインストーラが前記判断処理の結果を報知する。具体的には、システム内にインストール済のプリンタドライバの検索が行われ、インストール済のプリンタドライバのバージョンが古い場合には、既存のプリンタドライバの更新を行ってからでないと、プラグアンドプレイによるインストールは行われない旨のメッセージがホストコンピュータのCRT10に表示され（図2の②）、ドライバを更新するための手続きへと進む（図2の③）ようになっている。ここで、報知とは、ドライバインストーラ704を介してOSの機能を用いてホストコンピュータにダイアログボックスを用いて生成して表示することであってもよいし、ネットワークを介してドライバインストーラ704の遠隔操作を行っている場合などにおいては、上述したメッセージを含む電子メールを生成し、外部装置（図示省略）に対して送信することであってもよいし、音声を用いて上述のメッセージを通知することであってもよい。

【0038】図2の③はプログラム更新手段の好適な一例を示す。プログラム更新手段は、ドライバインストーラ704の一部として構成されてもよいが、OSの一部として組み込まれていてもよい。ユーザは図2の③において、既存のドライバを更新して、既存のプリンタドライバの整合性を保つべく、「ドライバを更新する」を選択して、図3の③のように、ドライバインストーラ704は、システム内のドライバのうち、新たにインストールされるプリンタドライバをインストールすると上書きされて整合性がとれなくなるプリンタドライバを更新する。さらに、必要に応じて装置を再起動して更新したブ

リンタドライバをOSに組ませる。この時、ユーザの指示、又は、装置の起動状態、又はUSBインターフェースから受信されるデバイスID情報を認識して、自動的にドライバインストーラ704が立ち上がるようにしておくと好適である。ドライバインストーラが起動されると、図2の①の画面が現れる。図2の①の画面の状態で「次へ」のボタンを押下すると、先ほどと同じく図5又は図6に記載された既にホストコンピュータに記憶されているプリンタAを制御するプリンタドライバが、不整合な状態となるか否かを判断する判断処理が行われるが、既にプリンタAについてはチェック済みの旨を検出し、既存のプリンタドライバモジュールはプリンタBのプリンタドライバモジュールを導入しても不整合な状態とならないと判断して、プラグアンドプレイインストーラの起動を許す。プラグアンドプレイインストーラは、自動的にデバイスID・ドライバパスのリスト701を読み出し、あるいは、デバイスID・ドライバパスのリスト701を指定して読み込ませることにより、プリンタAに対応するドライバパスを検出し、自動的にプリンタAに対応するドライバパスに配置されたプリンタドライバを取得し、ドライバ格納部706にプリンタドライバを格納すると共に、ドライバ管理部705にプリンタAを登録する。この時、ホストコンピュータ3000のシステム内のドライバは図3の④の状態となっており、プリンタA並びにプリンタBの両方について、バージョン2.0となって、プリンタドライバが正常にインストールされ、各プリンタドライバが整合性のとれた状態となっている。

【0039】なお、上記の実施例では、ドライバインストーラを起動させた後、図4の②で、USB接続のチェックボックスがチェックされるか否かの指示、すなわち、プラグアンドプレイインストーラを用いて、プラグアンドプレイインストールを行うか否かの指示により、プラグアンドプレイによるインストールが行われるか否かを識別している。しかし、より好適には、例えばUSBインターフェースを監視するプログラムを設けることにより、プラグアンドプレイインストールが行われることを、プラグアンドプレイインストーラが起動する前に検知し、当該受信したデバイスIDに対応するプリンタドライバに対応するプリンタドライバが導入されると、どのような不整合が起こりうるかの判断処理をしても良い。このように、制御手段の好適な一例であるドライバインストーラ704は、プラグアンドプレイインストーラ702により新しいプリンタドライバがインストールされた場合に既にホストコンピュータ3000に記憶されている既にインストールされているプリンタに対応するプリンタドライバが不整合な状態となるか否かを判断し、ホストコンピュータ3000のインストール動作を前記の判断結果に応じて制御することができる。装置のインストール動作の制御する方法の一例としては、ド

イバインストーラが704は判断結果に応じて、ホストコンピュータ3000の表示部の好適な一例であるCRT10を制御してダイアログボックスを表示させたり、プラグアンドプレイインストーラ702のインストール動作を制御したりすることが考え得る。また、ドライバインストーラ704が、判断結果に応じて、例えば判断結果を送受信するために、OSなどを介してホストコンピュータのLANやセントロなどのホストコンピュータ3000の一機能である通信インターフェース（入力部18）を間接的に制御して、ホストコンピュータ3000と他のコンピュータ等（図示省略）とを通信させることなども含む。

【0040】また、前記実施例については、既存のプリンタドライバが複数でない例を挙げていたが、例えば、既存のプリンタドライバが複数である場合には、ドライバインストーラに「全て更新」モードを設け、既存の各プリンタに対応するプリンタドライバのバージョンやインストールにともなうプログラムの整合性のチェックを先に行い、その後、更新すべき既存のプリンタドライバを特定し、ユーザの指示に応じて、又は、自動的に、まとめて更新させる処理をしてもよい。本実施形態によれば、プリンタドライバが不整合となるか否かを共有モジュールのバージョン情報並びに機種別モジュールのバージョン情報を用いて管理しているので、各モジュールのバージョン情報を比較するだけで、容易にプリンタドライバのインストールによって不整合となるか否かを判断できる。これらの処理のフローチャートについては、図5並びに図6を用いて詳述する。

【0041】図5は、第一実施形態におけるドライバインストーラのメインの処理を説明するための図であり、本実施例において、プラグアンドプレイによるドライバインストールの際に、既にインストール済みのプリンタドライバのモジュールが一部のみ更新されるのを防ぐため、既にインストール済みのプリンタドライバを更新してからでないと、プラグアンドプレイによるインストールが行われないようにした処理を示したフローチャートである。

【0042】以下、図5に示すフローチャートを中心にして本実施例を詳しく説明する。

【0043】本実施例における処理はユーザが図1に示したキーボードコントローラKBC5等により、プラグアンドプレイによるドライバのインストールをユーザが選択してドライバインストーラ704が起動することにより、又は、USBインターフェースに接続されたデバイスを認識するでドライバインストーラ704が自動的に起動することで始まる（S500）。

【0044】まず、S510およびS511の処理により、ドライバインストーラ704は、インストール済みのドライバの更新処理が必要かチェックを行っている。ドライバインストーラ704は、すべてのインストール

済みプリンタのチェックが終わっているかを判断し（S510）ドライバインストーラ704は、チェックが終わっていない場合には、S511の処理を行う。チェックが終わっている場合には、ドライバの更新が必要とされるプリンタのプリンタドライバを更新する必要がないという判断を行う。そして、S530でプラグアンドプレイインストーラが起動されてプラグアンドプレイインストール処理へ進む。S511では、ドライバインストーラ704は、プラグアンドプレイインストーラ702が新たにインストールしようとするプリンタドライバをインストールしたと仮定した場合に、S510で検出された、ドライバの更新が必要か否かの未チェックのプリンタに対応する既存のプリンタドライバの共通モジュールが上書きされるか否かの判断を行う。具体的には、ドライバインストーラ704は、インストール済みプリンタのプリンタドライバのファイル構成をチェックし、新規にインストールを行おうとしているプリンタドライバと共通のモジュール名がある場合には、そのバージョンを比較し、インストール済みのバージョンが古い場合には、モジュールが新しいもので上書きされると判断する。続いて、ドライバインストーラ704は、S512の処理を行う。それ以外の場合には、次のインストール済みのプリンタをチェックするため、ドライバインストーラ704は、再度S510の処理を行う。S512では、ドライバインストーラ704は、インストール済みのプリンタドライバを、新しいバージョンのプリンタドライバへ更新してからでないとプラグアンドプレイによるインストールはできない旨のメッセージを表示する。その後、ドライバインストーラ704による、プラグアンドプレイインストーラを介さないドライバの更新処理を行が行われ（S520）、処理を終了する。

【0045】以上のような処理により、プラグアンドプレイによるドライバインストールの際には、既にインストール済みのプリンタドライバは最新のものとなっているので、既にインストール済みのプリンタドライバの共通モジュールのみが上書きで更新されることにより発生する問題を回避することができる。

【0046】また、プリンタドライバの更新が必要な場合にはユーザに更新を促し、必要でない場合にはインストールを許可することで、ユーザ操作を補助すると共に、ドライバの整合性を保つことができる。

【0047】予め依存関係が整理された共有されたモジュール情報を管理する情報が入手可能である場合には、モジュール管理情報を活用することができるので、新たに追加するプリンタドライバをインストールすると共有モジュール情報が上書きされて整合性がとれなくなる可能性を容易に判断することが出来る。

【0048】さらに、ドライバインストーラ704を外部処理装置の接続に応答させて起動できるようにした場合には、外部処理装置の接続のタイミングに関わらず、

外部処理装置を接続してプラグアンドプレイが実行できる。

【0049】(第二実施形態) 図6は本実施例において、第二実施形態におけるドライバインストーラのメインの処理を説明するための図である。図6はプラグアンドプレイインストーラ702によるドライバインストールの際に、既にインストール済みのプリンタドライバのモジュールが一部のみ更新されるのを防ぐため、既にインストール済みのプリンタドライバを検出して更新してから、プラグアンドプレイによるインストールがまとめて行われるようにした処理を示したフローチャートである。

【0050】以下、第一実施形態との差分を示し、図6に示すフローチャートを中心にして本実施例を詳しく説明する。

【0051】本実施例における処理はユーザが図1に示したキーボードコントローラKBC5等により、プラグアンドプレイインストーラ702を使用したドライバのインストールをユーザが選択してドライバインストーラ704を起動させることにより、又は、USBインターフェースに接続されたデバイスを認識することでドライバインストーラ704が自動的に起動することで始まる(S600)。前記のプラグアンドプレイインストーラ702を用いたインストールを行う旨の選択指示は、例えば、図4の①に示したUSB接続でドライバをインストールするのチェックボックスを、キーボードコントローラKBC5又はホストコンピュータ3000に接続されたマウス(図示省略)からの入力によりチェックすることで行われるようにしてよい。まず、ドライバインストーラ704は、S610で未チェックの既存のプリンタがあるかをチェックする。ドライバインストーラ704は、S610で検出されたドライバの更新が必要であるプリンタに対して次の処理をS611で行う。検出は、USBのデバイスID情報を用いてもよいし、ドライバ管理部705に登録されているプリンタ情報を用いても良い。次に、ドライバインストーラ704は、外部から新しいプリンタドライバをインストールすると、ホストコンピュータ3000に既に記憶されたプリンタドライバが不整合な状態となるか否かをS611で判断する。具体的には、S611では、ドライバインストーラ704は、既存のプリンタドライバが、インストールしようとするプリンタドライバの新しいモジュールで上書きされて、既存のプリンタドライバの共有モジュールと機種別モジュールのバージョン情報の不一致な状態、すなわち、プリンタドライバが不整合な状態となるか否かを調べる。例えば、ドライバインストーラ704は、ドライバの更新が必要とされるプリンタのプリンタドライバが有する共通モジュール名を調べ、今回インストールしようとしているプリンタドライバが有する共通モジュール名を調べる。そして、ドライバインストーラ

704は、既存のプリンタが有するプリンタドライバのバージョンより新しい共通モジュールがあれば、ドライバの更新が必要とされるプリンタの共通モジュールが上書きされてプリンタドライバが不整合な状態となると判断し、ドライバの更新が必要か否かが未チェックのプリンタに対応するインストール済みのドライバの更新処理が必要と判断する。S611でドライバの更新が必要とされるか否かが未チェックのプリンタがプリンタドライバの共通モジュールが新しいモジュールで上書きされると判断された場合は、ドライバインストーラ704は、S620において、ドライバの更新が必要とされるプリンタのドライバの更新処理を行う。この際、図4の②に示した画面を表示し、OKボタンを押すと、図4の③に示したように、新しいプログラムのインストールを開始する前に、既存のプリンタドライバの共通モジュールが上書きされて不整合な状態とならないようすべく、ホストコンピュータ3000内の新たに追加されるプリンタドライバのインストールにより共通モジュールが上書きされると判断された既存のプリンタドライバを更新すべき旨を報知するドライバ画面をホストコンピュータ3000のCRT10に表示するようになると好適である。その後、システムインストーラ703によって、既存のドライバの更新が必要とされたプリンタドライバが更新される。そして、S610に戻って、ドライバインストーラ704は、未チェックのドライバの更新が必要とされるプリンタが存在しないかをチェックする。S610で全てのプリンタについて、プリンタドライバの更新が必要か否かのチェックが終わった場合は、プラグアンドプレイインストーラ702が、プラグアンドプレイによって検出されたプリンタに対応する新たなプリンタドライバのインストールを行って(S630)、インストール処理を終了する。

【0052】以上のような処理により、プラグアンドプレイによるドライバインストールの際には、既にインストール済のプリンタドライバは最新のものとなっているので、既にインストール済みのプリンタドライバの共通モジュールが上書きで更新されることにより発生する問題を回避することが可能となる。

【0053】さらに、一度モジュール更新のための作業を行った後、再度、新規に追加するプリンタのプリンタドライバをインストールする処理をユーザに促すことができ、例えばプリンタが動作しなくなったり、印刷システムの不具合を引き起こしたりするような、ユーザの意図しない障害を引き起こす可能性のある制御プログラムへの変更を防ぐことが出来る。また、共通モジュールを含むプリンタドライバを更新する場合は、モジュールの依存関係を示す共通モジュール管理情報が利用できるので、簡単に新たにプリンタドライバを追加更新した場合に、整合性が取れるか否かを事前にチェックできると共に、必要な場合には既存のプリンタドライバを更新した

上で、目的のプリンタドライバをプラグアンドプレイによりインストールすることが出来る。

【0054】なお、最初に更新する必要があるプリンタドライバを特定し、まとめて該プリンタドライバを更新することで、インストーラへの指示が少なくてすみ、プリンタドライバ更新時のユーザーの操作が簡略化できるという効果がある。

【0055】なお、本発明は、複数の機器(例えばホストコンピュータ、インターフェース機器、リーダ、プリンタなど)から構成されるシステムに適用しても、一つの機器からなる装置(複写機、プリンタ、ファクシミリ装置など)に適用してもよい。また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記憶した記録媒体を、システムあるいは装置のコンピュータ(またはCPUやMPU)が記録媒体に格納されたプログラムコードを読み出し、実行することによっても達成される。

【0056】この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記録媒体は本発明を構成することになる。

【0057】プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0058】また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼動しているOS(オペレーティングシステム)などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。システムを管理するソフトウェアの好適な一例としてOSを挙げたが、OS以外のシステムソフトウェアであってもよい。また、プログラム導入手段がシステムソフトウェアに含まれる場合でも、本発明の目的を達成するドライバインストーラの機能を含むプログラムは、本発明の範疇に含まれることは言うまでもない。また、新規又は更新インストールによって既存プリンタドライバが不整合となるか否かの判断処理は、前記インストールの前に行うのが効果的であるが、例えば、ホストコンピュータ内の既存ドライバを退避させるなどして、新規又は更新インストールを行ってしまってから、前記の判断処理を行い、判断処理の結果に応じて既存ドライバを元に戻すか否かを決定するようにしてもよい。

【0059】さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示

に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0060】以上説明したように、本発明の上記実施形態によれば、プラグアンドプレイによるインストール時には、インストール済みのプリンタドライバのモジュールファイルは常にすべて新しいモジュールファイルにより更新されていることとなり、一部のモジュールファイルのみが新しいバージョンのファイルに書き換えられてしまうといった状況を回避でき、新たにプリンタドライバがインストールされたことにより、プリンタドライバが不整合となることによって既にインストールされていたプリンタドライバの動作が不正になるのをさけることが可能となる。

【0061】

【発明の効果】以上説明したように、本願発明の実施形態によれば、プラグアンドプレイインストーラを用いる場合であっても、他の既存の制御プログラムが整合性を保つようにインストールすることができる。また、本願の実施形態の一つによれば、外部処理装置が接続されたことに応答して、外部処理装置の識別情報を受信し、前記外部処理装置に対応する第一の制御プログラムを、情報処理装置にインストールする、プログラム導入手段を有する情報処理装置において、前記第一の制御プログラムをインストールすると、既に前記情報処理装置に記憶された前記外部処理装置の第二の制御プログラムが不整合な状態となるか否かを判断する判断手段を有することを特徴としたので、プラグアンドプレイインストーラを用いて新たに制御プログラムのインストールを行う場合であっても、他の既存の制御プログラムが、インストールされるプログラムによって意図しない上書きをされることなく、整合性を保つことができるようになつた。

【図面の簡単な説明】

【図1】本発明の実施形態における制御プログラムのインストールシステムの実施例を示すプリンタ制御システムの構成を示す図である。

【図2】情報処理装置であるホストコンピュータ3000における典型的な印刷処理の例を示す図である。

【図3】ドライバを構成するモジュールの内容、並びに、新たにドライバをインストールする際の処理を示す流れ図である。

【図4】本実施例におけるドライバインストーラ704を起動した際に表示されるプリンタドライバのインストール処理、並びに、インストール処理を選択するためのユーザインターフェースの例を示す図である。

【図5】第一実施形態におけるドライバインストーラのメインの処理を説明するための図である。

【図6】第二実施形態におけるドライバインストーラの

メインの処理を説明するための図である。

【図7】本発明の実施例におけるホストコンピュータ3

000内のソフトウェア構成を示す図である。

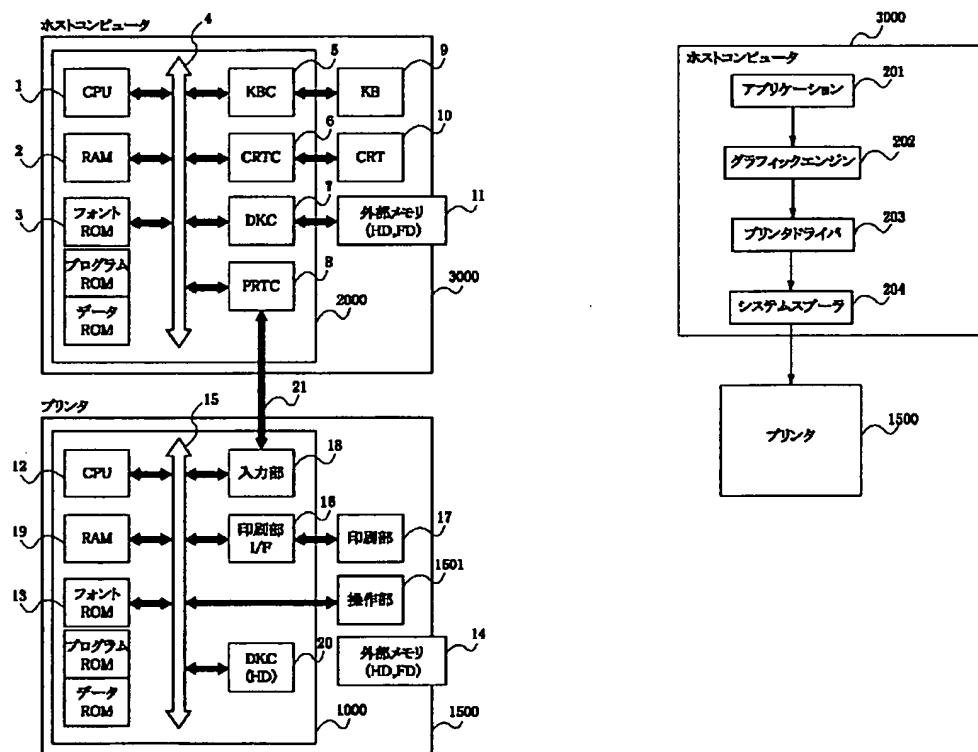
【符号の説明】

- 1 CPU
- 2 RAM
- 3 ROM
- 4 システムバス
- 5 キーボードコントローラ (KBC)
- 6 CRTコントローラ (CRTC)
- 7 ディスクコントローラ (DKC)
- 8 プリンタコントローラ (PRTC)
- 9 キーボード (KB)
- 10 CRT
- 11 外部メモリ (HD, FD)
- 12 CPU
- 13 ROM

- 1 4 外部メモリ (HD, FD)
- 1 5 システムバス
- 1 6 印刷部 I/F
- 1 7 印刷部
- 1 8 入力部
- 1 9 RAM
- 2 0 ディスクコントローラ・DKC (HD)
- 1 5 0 0 プリンタ
- 1 5 0 1 操作部
- 3 0 0 0 ホストコンピュータ
- 7 0 1 デバイス ID・ドライバリスト
- 7 0 2 プラグアンドプレイインストーラ
- 7 0 3 従来のインストーラ (システムインストーラ)
- 7 0 4 ドライバインストーラ
- 7 0 5 ドライバ管理部
- 7 0 6 ドライバ格納部

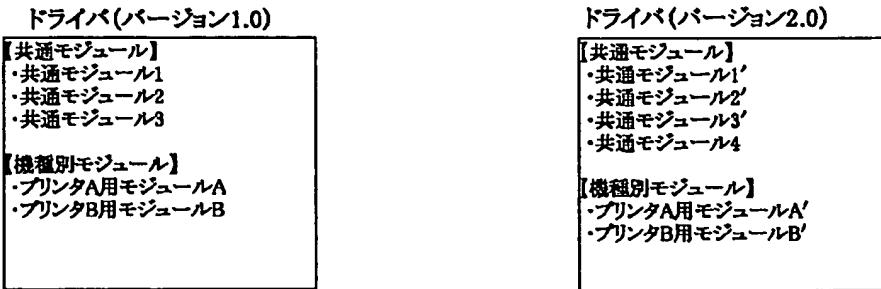
[図1]

[图2]

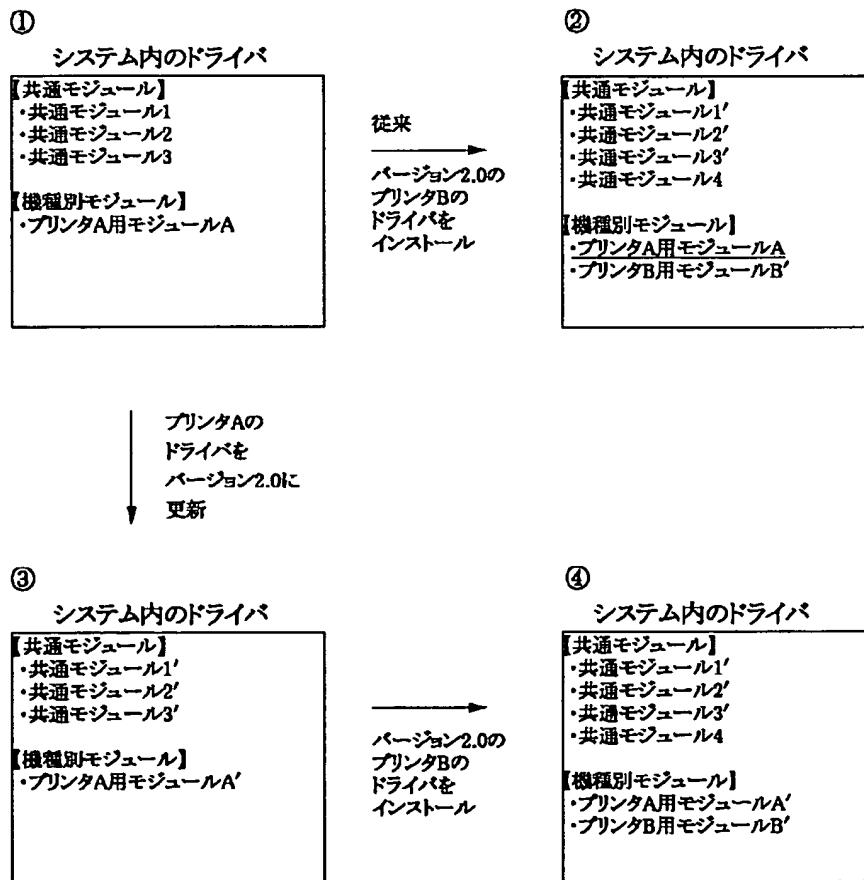


【図3】

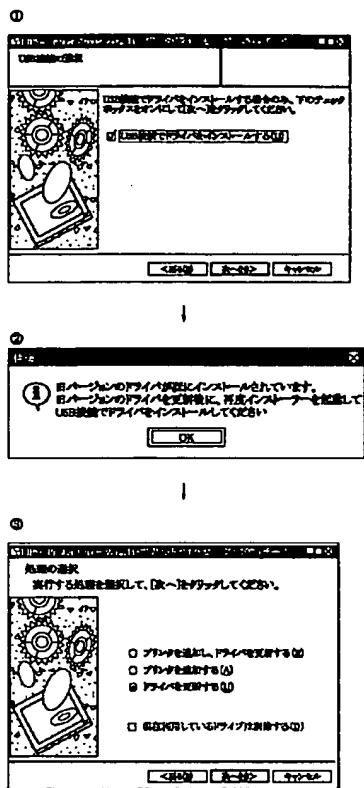
ドライバを構成するモジュール



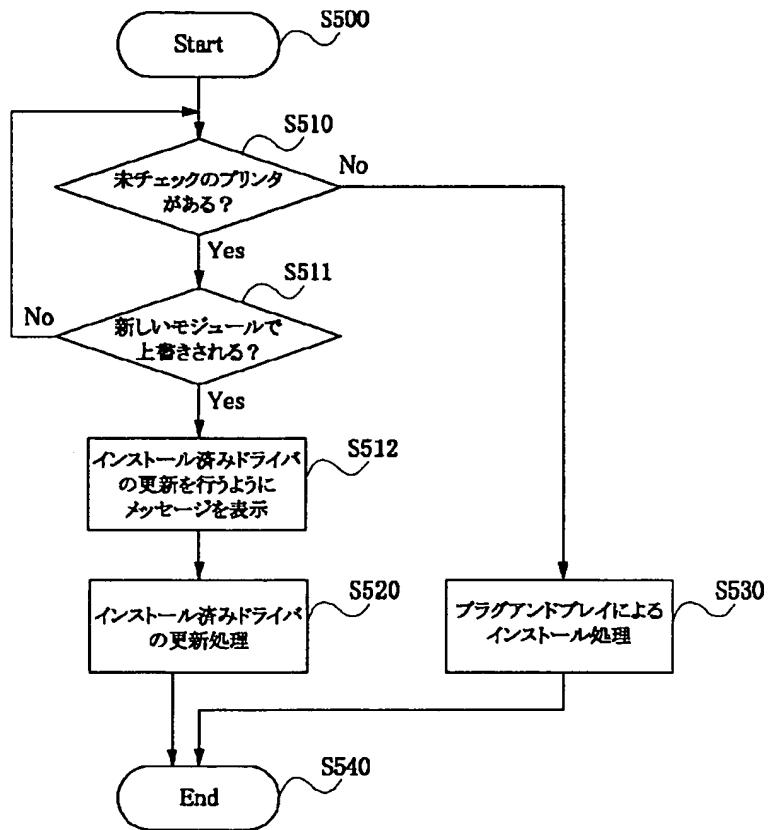
プリンタBをプラグアンドプレイで追加した場合の流れ図



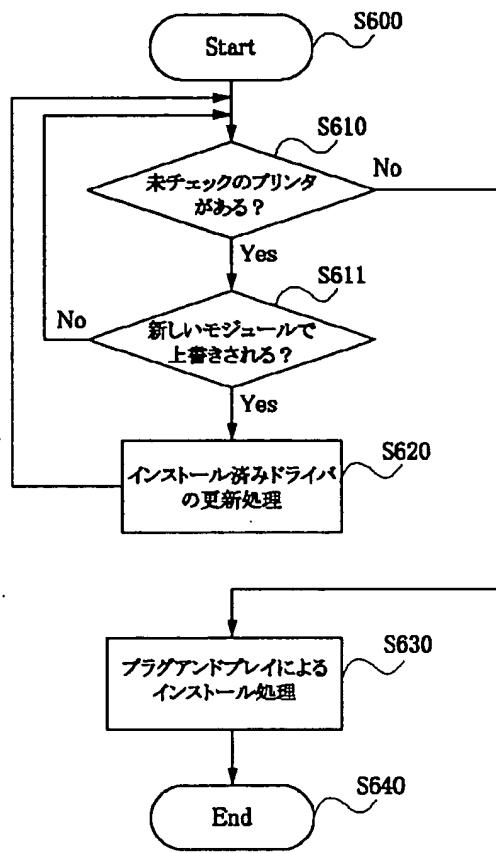
【図4】



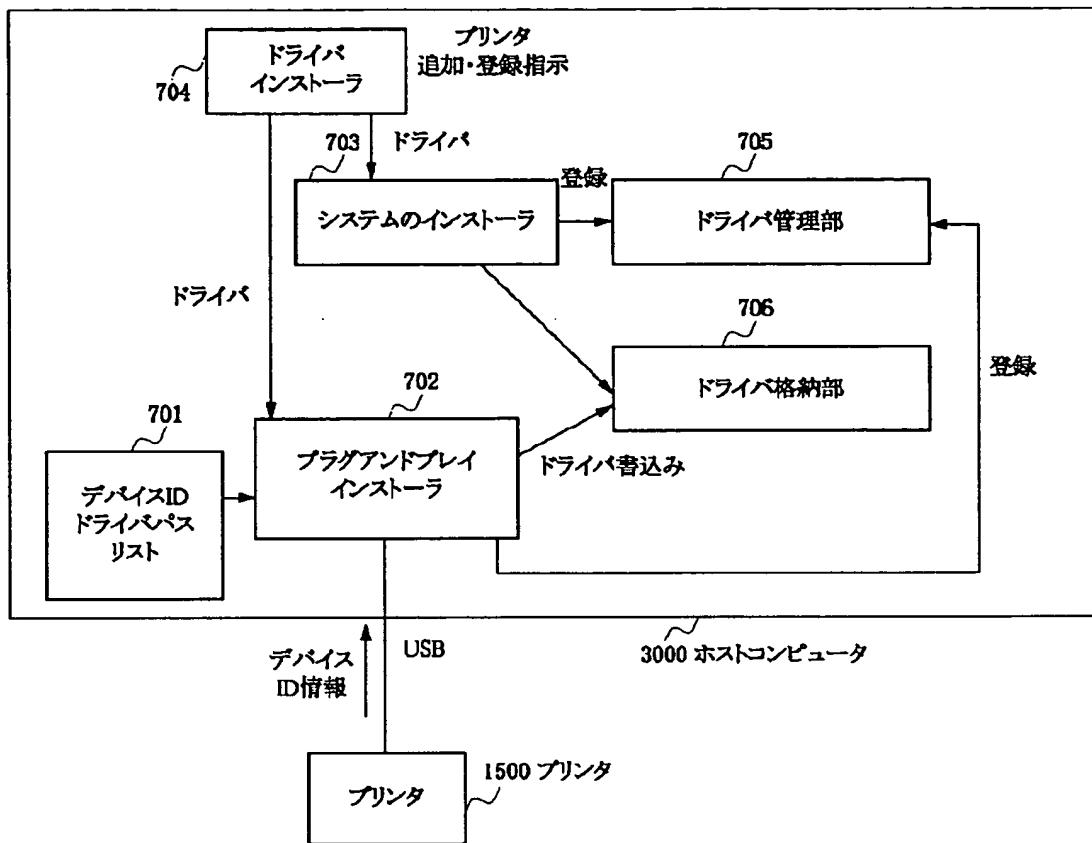
【図5】



【図6】



【図7】



【手続補正書】

【提出日】平成14年12月17日(2002.12.17)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】0037

【補正方法】変更

【補正内容】

【0037】例えば、今、ホストコンピュータ3000のシステム内のドライバが、図3の①の状態になつてゐる。まず、ユーザはプリンタドライバインストーラを起動し、プラグアンドプレイ機能によってプリンタドライバをインストールするために、「USB接続でドライバをインストールする」をチェックする(図4の①)。その後、「次へ」ボタンを押下すると、後に図5並びに図6を用いて詳述するが、プリンタBを制御するプリンタドライバをインストールすると、既にホストコンピュータに記憶されているプリンタAを制御するプリンタドライバが、不整合な状態となるか否かを判断する判断処理(図5のS510及びS511、又は、図6の

S610又はS611)が行われ、プラグアンドプレイインストーラ702によるプリンタBのプリンタドライバのインストールが開始される前に、ドライバインストーラが前記判断処理の結果を報知する。具体的には、システム内にインストール済のプリンタドライバの検索が行われ、インストール済のプリンタドライバのバージョンが古い場合には、既存のプリンタドライバの更新を行つてからでないと、プラグアンドプレイによるインストールは行われない旨のメッセージがホストコンピュータのCRT10に表示され(図4の②)、ドライバを更新するための手続きへと進む(図4の③)ようになつてゐる。ここで、報知とは、ドライバインストーラ704を介してOSの機能を用いてホストコンピュータにダイアログボックスを用いて生成して表示することであつてもよいし、ネットワークを介してドライバインストーラ704の遠隔操作を行つてゐる場合などにおいては、上述したメッセージを含む電子メールを生成し、外部装置(図示省略)に対して送信することであつてもよいし、音声を用いて上述のメッセージを通知することであつて

もよい。

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0038

【補正方法】変更

【補正内容】

【0038】図4の③はプログラム更新手段の好適な一例を示す。プログラム更新手段は、ドライバインストーラ704の一部として構成されてもよいが、OSの一部として組み込まれていてもよい。ユーザは図4の③において、既存のドライバを更新して、既存のプリンタドライバの整合性を保つべく、「ドライバを更新する」を選択して、図3の③のように、ドライバインストーラ704は、システム内のドライバのうち、新たにインストールされるプリンタドライバをインストールすると上書きされて整合性がとれなくなるプリンタドライバを更新する。さらに、必要に応じて装置を再起動して更新したプリンタドライバをOSに組込ませる。この時、ユーザの指示、又は、装置の起動状態、又はUSBインターフェースから受信されるデバイスID情報を認識して、自動的にドライバインストーラ704が立ち上がるようにしておくと好適である。ドライバインストーラが起動されると、図4の①の画面が現れる。図4の①の画面の状態

で「次へ」のボタンを押下すると、先ほどと同じく図5又は図6に記載された既にホストコンピュータに記憶されているプリンタAを制御するプリンタドライバが、不整合な状態となるか否かを判断する判断処理が行われるが、既にプリンタAについてはチェック済みの旨を検出し、既存のプリンタドライバモジュールはプリンタBのプリンタドライバモジュールを導入しても不整合な状態とならないと判断して、プラグアンドプレイインストーラの起動を許す。プラグアンドプレイインストーラは、自動的にデバイスID・ドライバパスのリスト701を読み出し、あるいは、デバイスID・ドライバパスのリスト701を指定して読み込ませることにより、プリンタAに対応するドライバパスを検出し、自動的にプリンタAに対応するドライバパスに配置されたプリンタドライバを取得し、ドライバ格納部706にプリンタドライバを格納すると共に、ドライバ管理部705にプリンタAを登録する。この時、ホストコンピュータ3000のシステム内のドライバは図3の④の状態となっており、プリンタA並びにプリンタBの両方について、バージョン2.0となって、プリンタドライバが正常にインストールされ、各プリンタドライバが整合性のとれた状態となっている。